SPLICE JUNCTION CLASSIFICATION PROBLEMS FOR DNA SEQUENCES: REPRESENTATION ISSUES

Manish Sarkar and Tze-Yun Leong
Department of Computer Science, School of Computing, The National University of Singapore, Singapore: 119260
E-mail: {manish, leongty}@comp.nus.edu.sg

Abstract- **Splice junction classification in a Eukaryotic cell is an important problem because the splice junction indicates which part of the DNA sequence carries protein-coding information. The major issue in building a classifier for this classification task is how to represent the DNA sequence on computers since the accuracy of any classification technique critically hinges on the adopted representation. This paper presents the experimental results on seven representation schemes. The first three representations interpret each DNA sequence as a series of symbols. The fourth and fifth representations consider the sequence as a series of real numbers. Moreover, the first, second and fourth representations do not consider the influence of the neighbors on the occurrence of a nucleotide, whereas the third and fifth representations take the influence of the neighbors into considerations. To capture certain regularity in the apparent randomness in the DNA sequence, the sixth representation treats the sequence as a variant of random walk. The seventh representation uses Hurst coefficient, which quantifies the roughness of the DNA sequence. The experimental results suggest that the fourth representation scheme makes sequences from the same class close and the sequences from the different classes far, and thus finds a structure in the input space to provide the best classification result.**

Keywords - **Gene, DNA, exon, intron, representation, splice boundary, classification, random walk and Hurst coefficient.**

I. INTRODUCTION

Problem description: The biochemical material that carries hereditary characteristics from parents to offspring is contained in a sequence of chemical known as *deoxyribonucleic acid* (DNA). A *gene* consists of a continuous stretch of DNA that is needed to produce a particular protein. The process by which the DNA gives rise to a protein is called *gene expression*. In a eukaryotic cell, i.e., the cell that contains a nucleus, the gene expression involves the synthesis of premRNA on the DNA templates (*transcription*), removal of the non-coding region (*splicing*) from the premRNA to form mRNA, and the synthesis of the protein on the mRNA templates (*translation*). Due to the splicing, a DNA sequence consists of alternating segments of *exon* and *intron*, where an exon is a nucleotide sequence that is expressed or translated into protein, and an intron is an intervening sequence that is transcribed into RNA, but later eliminated from the transcript by splicing its adjacent exons (Fig. 1). The splice junction refers to the point where the splicing takes place, i.e., it is the meeting point of intron and exon.

Motivation: Localization of protein coding region in a DNA sequence by pure biological means is a time-consuming and costly procedure. Hence, many computational methods have been attempted to recognize the splice junctions. To the best of our knowledge, no paper has highlighted on studying the representation issues of the splice junction problem. The performance of any classification technique critically depends on the representation. For instance, the DNA sequence can be represented as a symbol, binary numbers or real numbers. In addition, we can exploit the influence of the neighbors or some other characteristics of the sequence to form better representation schemes.

Objective: This work conducts experiments to find the appropriate representation for the splice junction classification task. It involves (a) recognizing exon/intron boundaries, or “donor” sites, (b) recognizing intron/exon boundaries, or “acceptor” sites, and (c) neither. To measure the generalization capability of the classifier, we have used standard classification error measures.

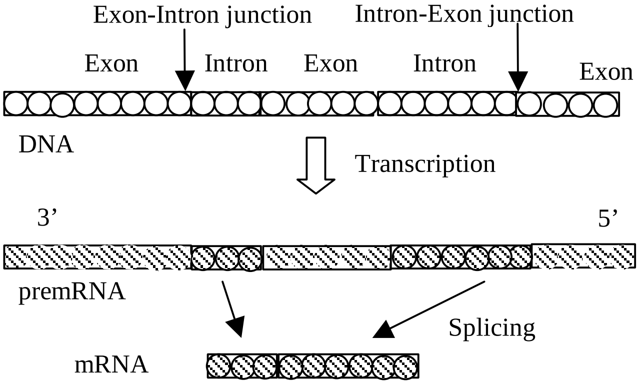

Fig. 1: *The transcription and spicing of a gene in the nuclease of a eukaryotic cell. Each circle in the DNA sequence represents a nucleotide, i.e., any one of Adenine (A), Thiamine (T), Cytosine (C) or Guanine (G). Each circle in the premRNA and mRNA is A, U (Urethane), C or G.*

Scope: The data set, which we have collected from [Blake:98], contains 3190 samples. Approximately 25% samples of the data set have intron-exon boundaries, the other 25% samples have exon-intron boundaries, and the remaining 50% samples have no boundaries. Each sample is a sequence of 60 nucleotides, which we denote by F1 through F60, and the boundary (if any) is just at the middle (Fig. 2). Each sequence is any one from the three classes, and the aim is to identify the midpoint of the sequence as being an exon-intron (EI) boundary, an intron-exon (IE) boundary, or neither boundary (N).

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
25OCT2001	N/A	-

Title and Subtitle
Splice Junction Classification Problems for DNA Sequences: Representation Issues

Contract Number

Grant Number

Program Element Number

Author(s)

Project Number

Task Number

Work Unit Number

Performing Organization Name(s) and Address(es)
Department of Computer Science, School of Computing, The National University of Singapore, Singapore: 119260

Performing Organization Report Number

Sponsoring/Monitoring Agency Name(s) and Address(es)
US Army Research, Development & Standardization Group (UK) PSC 802 Box 15 FPO AE 09499-1500

Sponsor/Monitor's Acronym(s)

Sponsor/Monitor's Report Number(s)

Distribution/Availability Statement
Approved for public release, distribution unlimited

Supplementary Notes
Papers from the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, October 25-28, 2001, held in Istanbul, Turkey. See also ADM001351 for entire conference on cd-rom., The original document contains color images.

Abstract

Subject Terms

Report Classification
unclassified

Classification of this page
unclassified

Classification of Abstract
unclassified

Limitation of Abstract
UU

Number of Pages
4

Issues: The aim of an appropriate representation is to form a structure in the input space such that two sequences from the same class remain in the close vicinity in the input space, and the sequences from different classes remain wide apart. This problem could be posed by considering the frequency of occurrence of the nucleotides or by defining certain distance function in the input space. Both of these two techniques implicitly compress the information needed to represent the input sequence. It is achieved by making the representations similar (dissimilar) for sequences of same (different) class. The compression becomes more effective when some intrinsic properties of the sequence like influence of the neighbors, roughness are reflected in the representation.

Another major problem is the representation of the sequences that have neither exon/intron nor intron/exon boundary. Representing patterns for this class is difficult since (a) the space covered by this class is very large, and often not enough training patterns are present to cover such a large space, and (b) this class and the other two classes are overlapping.

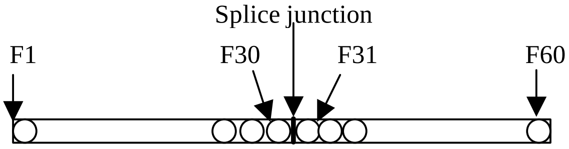

Fig. 2: *Splice junction classification problem with the given set of data. Each DNA sequence contains sixty nucleotides, and their position is indicated from the left by F1, F2,…, F60. The boundary (if any) occurs between F30 and F31.*

II. Methodology

Representation 1: A pure symbolic approach is adopted in this representation. All the four possible nucleotides are represented using different symbols, for instance *A* for Adenine, *T* for Thiamine, *C* for Cytosine and *G* for Guanine. Hence, a sequence of length 60 is represented as series of 60 symbols. We can construct a rule base of all possible (4^{60}) *if-then* rules that relate any valid sequence of length 60 and the output class. When a new sequence arises, the class label of this sequence can be determined using the class label of the matching sequence in the rule base. However, constructing and accessing such a large rule base is computationally intractable. In contrast, if we construct a database with less number of rules, then for some valid input sequence, no rule may fire because there is no match between the input and the rules. Hence, we need to decrease the number of rules without compromising much in the classification efficiency. In other words, from a small set of rules the classifier needs to generalize such that the classifier can classify any sequence with high classification efficiency. The generalization with less number of rules is possible if some property of the sequence is reflected in the representation so that it can be subsequently captured by the classifier. To achieve that, we separate the given DNA sequences into two sets: training set *Tr* and testing set *Ts* ($Tr \cap Ts = \varnothing$). The training set *Tr* is used for building the classifier, and the testing set *Ts* is used to determine how efficient the resultant classifier is. The resultant classifier can be used to classify any valid input sequence. Below we describe some approaches along this line.

Representation 2: It has been observed that inside intron, not all triplets of nucleotides (called *codons*) appear with the same probability. Specifically, the probability of occurrence of a nucleotide in intron is different for each position. The exon does not have this property. Hence, this property can be exploited to find the difference between intron region and exon region. Each position in the codon is represented by 0, 1, and 2. Hence any nucleotide can be viewed as a member of the alphabet {*A0, A1, A2, T0, T1, T2, C0, C1, C2, G0, G1, G2*}. *A1* indicates that the nucleotide is Adenine and it at the second position of the codon. Using this representation, the *Jenson-Shanon divergence measure* [Galvan:00] is computed for the two halves (F1 to F30 and F31 to F60). The Jenson-Shanon divergence measure is supposed to attain the maximum value at the point where two dissimilar regions are merged. If the divergence measure crosses some threshold, then it is considered as the splice junction. The appropriate value of the threshold is estimated from the training set.

Representation 3: Like the first representation, four different symbols are used here. Here the focus is on (a) the frequency of occurrence of a particular nucleotide at a particular position, and (b) how the occurrence of that nucleotide is influenced by the previous nucleotides in the series. As a classifier, we have used hidden Markov model with five states. We train the model using the Baum-Welch algorithm [Bengio:99] on the training set.

Representation 4: Here we are interested in the structure of the input space. In this representation, each nucleotide is represented as [Towel:94]

$$A = 0001,\ G = 0010,\ C = 0100 \ \text{and}\ T = 1000 \qquad (1)$$

Note that here each nucleotide is occupying a corner of a four dimensional hypercube, and hence the distance between any two nucleotides is constant. Another possible representation is *A* = 00, *T* = 01, *C* = 10, *G* = 11. But this representation is more biased from the Euclidean distance sense since it indicates *A* is closer to *T* than *G*. Hence, we have adopted the representation of Equation (1). The sequence *ATC* is represented as 0001 1000 0100. Thus, each DNA sequence of the training and test set is represented as a string of 240 zeros and ones.

We have applied a feedforward neural network with backpropagation learning as a classifier [Jang:97]. The network has three layers, and the numbers of nodes in the first, second and third layers are 240, 10 and 3, respectively.

The classifier searches 240-dimensional hyperspace to find a structure in the hyperspace. The classifier classifies a test sequence based on in which structure the test pattern falls. Note that while forming the input space or while searching the input space, we do not consider explicitly the interaction between two neighboring nucleotides.

Representation 5: In this representation, we express a DNA sequence using the same format as in Equation (1). But, the sequence is formed as a 4×60 vector. Hence, the sequence *ATC* is represented as

$$\begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{matrix} \qquad (2)$$

In addition, the sequence is considered similar to a time series. It means the appearance of a nucleotide depends on the previous nucleotide of the series.

We have constructed a time delay feedforward neural network (with four unit time delay) that approximates the sequences of Tr for a particular class [Plataniotis:96]. For three classes we have used three neural networks of same configurations: 20, 5 and 4 nodes in the first, hidden and output layers, respectively. The networks act as predictors, and their prediction error is used to train them. Let us assume a sequence $\boldsymbol{x} = [x_1 x_2 \ldots x_{60}]' \in Tr$ is from the class EI. When a part of the sequence $\boldsymbol{x}$ (say $x_1 x_2 \ldots x_{15}$) is fed, the network predicts (say o_{16}) the nucleotide at the 16th position based on the last four nucleotides (i.e. $x_{12} x_{13} x_{14} x_{15}$) in $\boldsymbol{x}$. Now the difference between o_{16} and x_{16} is found out. This difference acts as an error to train the network iteratively so that the prediction becomes more accurate. Next the difference between the network output o_{17} and the actual nucleotide value x_{17} is calculated. Again, the difference is used to train the network. This procedure is carried out for the whole sequence $\boldsymbol{x}$, and it is repeated for all the sequences of the given class. Thus for the three classes, three predictors are trained.

When a new sequence $\tilde{\boldsymbol{x}} = [\tilde{x}_1 \tilde{x}_2 \ldots \tilde{x}_{60}]' \in Ts$ appears, it is fed to all the predictors. Initially $\tilde{x}_1 \tilde{x}_2 \tilde{x}_3 \tilde{x}_4$ is fed to the predictor for the class EI. It produces the output o_5. Now the error is computed as $e_5 = (\tilde{x}_5 - o_5)^2$. Following the similar procedure, the total error by the predictor is $\grave{o}_{EI} = \sum_{i=5}^{60} (\tilde{x}_i - o_i)^2$. Similarly the errors for the predictors corresponding to the other classes are calculated. The predictor that produces the least error is the model closest to the sequence, and hence the class label of the closest predictor is accepted as the class label of the test sequence.

Representation 6: This measure attempts to extract some regularity from the apparent randomness inherent in the DNA sequence. In the conventional random walk model, a walker moves either *up* ($u_i = +1$) or *down* ($u_i = -1$) by one unit length at the *i*th step of the walk. Following this concept, one definition of the DNA walk is that the walker steps *up* if a pyrimidine (*C* or *T*) occurs at the position *i* along the DNA chain, while walker steps *down* if a purine (*A* or *G*) occurs at the *i*th position [Havlin:99]. Thus at the *i*th nucleotide the position of the walker is $Y_i = \sum_{j=1}^{i} u_j$, and the sequence appears as a time series. Note that in the Representation 4 and 5, the sequence can have values only in {0, 1}; but now the sequence can have any positive or negative discrete value. The classification is carried out in the following steps: (a) Represent all sequences in the training and test sets with the DNA walk representation. (b) Find the mean sequences for each class. For instance, the mean sequence for the class EI is

$$m_{EIj} = \frac{\sum_{\boldsymbol{Y}_i \text{ is from } Tr \text{ and class EI}} Y_{ij}}{\text{no. of sequences in } Tr \text{ that are from class } EI}$$

(c) When a new test sequence $\tilde{\boldsymbol{Y}} = [\tilde{Y}_1 \tilde{Y}_2 \ldots \tilde{Y}_{60}]'$ appears, we find the mean sequence closest to it using the following similarity measure: $S_{EI}(\boldsymbol{m}_{EI}, \tilde{\boldsymbol{Y}}) = \sum_{j=1}^{60} (m_{EIj} - \tilde{Y}_j)^2$. (d) The class label of $\tilde{\boldsymbol{Y}}$ is the class label of the closest mean sequence.

Using the DNA walk representation, we have plotted the traces corresponding to all the three classes (Fig. **3**). We can observe that to some extent the lines representing the classes IE and EI can be separated visually; however, even for a human observer it is difficult to separate the lines corresponding to the class N from the lines of the other two classes. It shows that no classifier with the DNA walk representation can produce high classification efficiency for the class N.

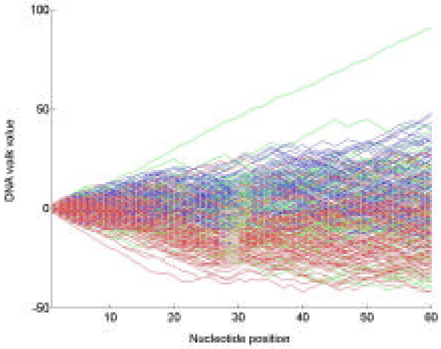

Fig. 3: The blue, red and green lines indicate the DNA walk for the genes with intron/exon boundary, exon/intron boundary and neither. We can observe that blue and red lines can be separated better than the red and green, and blue and green lines. Hence, while using the DNA walk representation, the classifiers also cannot classify the sequences of the class N.

Representation 7: In the DNA walk, it has been noticed that if the ruggedness or irregularity of a part of the DNA walk is scaled up, then the resultant ruggedness becomes similar to the ruggedness or irregularity of the whole sequence [Havlin:99]. This clue can act as the regularity in the DNA walk, and thus it can be used to characterize the time series. The *Hurst exponent* intends to quantify this clue such that the quantified values are relatively insensitive to translation, scaling, noise and nonstationarity [Addision:97]. One popular approach to estimate the Hurst exponent is the *dispersional analysis*. It needs the following three steps to be performed on a sequence:

1. *Partitioning:* Each sequence is partitioned into equal intervals of length l. Let us call the *i*th interval $W(i, l)$.

2. *Single scale statistics:* It can be of two types:

(a) *Local statistics:* Statistics based on the values of the DNA walk within a single interval are extracted. The local statistic in the interval $W(i, l)$ is the mean of the data in each interval.

(b) *Partition based statistics:* The partition-based statistic $I(l)$ is the standard deviation of the means. Next, the whole process is repeated for several lengths of the interval l.

3. *Transscale statistics:* The transscale statistics is 1.0 plus the slope of a linear regression that fits a plot of $\log(I(l))$ vs. $\log(l)$ for all *l*.

The Hurst exponents are extracted for exon and intron regions of each sequence in the training set. The extracted Hurst coefficients are used to train a two-class feedforward neural network with backpropagaton learning. Whenever a new sequence appears, the Hurst coefficients of its two halves are determined, and then the Hurst exponent of each half is fed to the neural network to identify the type of that half. The splice junction can be classified easily after we know the identity of each half.

III. Results and discussion

In the data set, we have removed the sequences where some nucleotides have unknown values. Half of the available sequences are used for training and the remaining half are used for testing. We have not studied Representation 1 as it is computationally intractable. The classification performances using the remaining six representations are shown in TABLE 1. We can observe that the fourth representation is providing the best result even for the class N. This conclusion remains valid even when we performed experiments on three other large data sets obtained from GenBank. It indicates that the structure of the input space is important, Representation 4 is able to form a better structure in the input space and the dependency of the neighbors is implicitly captured in Representation 4.

TABLE 1

COMPARATIVE CLASSIFICATION PERFORMANCE WITH SIX REPRESENTATIONS. THE CLASSES ARE INTRON-EXON (IE) BOUNDARY, EXON-INTRON (EI) BOUNDARY AND NEITHER (N).

Representation	IE	EI	N	Overall
2	34.34%	46.32%	61.32%	50.82%
3	72.45%	78.56%	45.67%	60.58%
4	85.16%	93.12%	70.45%	79.79%
5	67.23%	76.35%	22.45%	47.12%
6	82.67%	91.47%	9.32%	48.19%
7	52.45%	54.37%	54.37%	53.89%

Note that (a) instead of neural networks or hidden Markov models, other classifiers could be used, and in that case the classification performances may vary, (b) for Representation 1, 4, 5 and 6, all training and testing sequences should be of same length, although for Representation 2, 3 and 7 this constraint is not present, and (c) due to the space limitation, we could not discuss many other representations.

The future works would be (a) construction of a better classifier using Representation 4, (b) extraction of classification rules from the data set, (c) studying the efficiency of the Representation 4 when some DNA sequence has missing values, and (d) developing a modular approach involving all the representations.

References

[Addision:97] P. A. Addison. *Fractals and Chaos: An Illustrated Course.* Institute of Physics Publishing, London, 1997.

[Benigo:99] Y. Bengio. Markovian models for sequential data. *Neural Computing Surveys*, vol. 2, pp. 129-162, 1999.

[Blake:98] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, http://www.ics.uci.edu/~mlearn, 1998.

[Galvan: 00] P. B. Galvan, I. Grosse, C. Pedro, J. L. Oliver, R. R. Roldan and H. E. Stanley. Finding borders between coding and noncoding DNA regions by an entropic segmentation method, *Physical Review Letters*, Vol. 85, No. 6, August 2000, pp. 1342-1345

[Havlin:99] S. Havlin, S. V. Buldyrev, A. Bunde, A. L. Goldberger, P. C. Ivanov, C. K. Peng and H. E. Stanley. Scaling in nature: From DNA through heartbeats to weather. *Physica A*, no. 273, pp. 46-69, 1999.

[Jang: 97] J. S. R. Jang and C. T. Sun and E. Mijutani. *Neuro-Fuzzy and Soft Computing*, Prentice Hall, Englewood Cliffs, NJ, 1997.

[Plataniotis:96] K. N. Plataniotis, D. Androutsos, A. N. Venetsanopoulos and D. G. Lainiotis. A new time series classification approach. Signal Processing, no. 54, pp. 191-199, 1996.

[Towel:94] G. G. Towell and J. W. Shavlik, Knowledge based artificial neural networks, *Artificial Intelligence*, vol. 70, no. 1-2, 1994, pp. 119-165.